

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
22 November 2001 (22.11.2001)

PCT

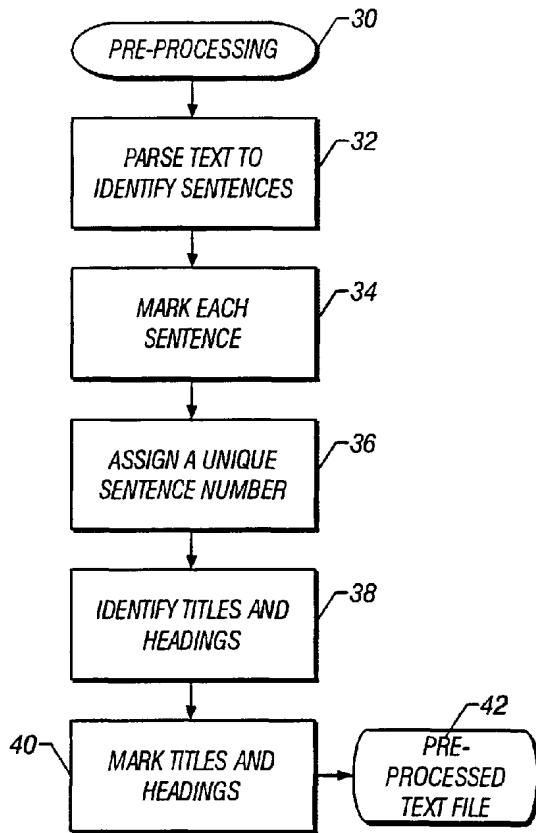
(10) International Publication Number
WO 01/88662 A2

(51) International Patent Classification ⁷ :	G06F	09/637,616	11 August 2000 (11.08.2000)	US
(21) International Application Number:	PCT/US01/15711	(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier applications:		
(22) International Filing Date:	16 May 2001 (16.05.2001)	US	09/572,770 (CON)	
(25) Filing Language:	English	Filed on	17 May 2000 (17.05.2000)	
(26) Publication Language:	English	US	09/573,025 (CON)	
(30) Priority Data:		Filed on	17 May 2000 (17.05.2000)	
09/572,770	17 May 2000 (17.05.2000)	US	09/572,276 (CON)	
09/573,025	17 May 2000 (17.05.2000)	Filed on	17 May 2000 (17.05.2000)	
09/572,276	17 May 2000 (17.05.2000)	US	09/573,024 (CON)	
09/572,186	17 May 2000 (17.05.2000)	Filed on	17 May 2000 (17.05.2000)	
09/573,023	17 May 2000 (17.05.2000)	US	09/637,616 (CON)	
09/573,024	17 May 2000 (17.05.2000)	Filed on	11 August 2000 (11.08.2000)	

[Continued on next page]

(54) Title: ANSWERING NATURAL LANGUAGE QUERIES

(57) Abstract: Natural language query systems and applications are described.



WO 01/88662 A2



(71) **Applicant** (*for all designated States except US*): ANSWERFRIEND.COM [US/US]; 800 West Sixth Street, #1000, Los Angeles, CA 90017 (US).

(72) **Inventors; and**

(75) **Inventors/Applicants** (*for US only*): MEKIKIAN, Gary [US/US]; 115 Skyline Drive, Burbank, CA 91501 (US). YURET, Deniz [TR/US]; 310 The Village, #117, Redondo Beach, CA 90277 (US).

(74) **Agent:** FEIGENBAUM, David, L.; Fish & Richardson, P.C., 225 Franklin Street, Boston, MA 02110-2804 (US).

(81) **Designated States (national):** AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW,

MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) **Designated States (regional):** ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

ANSWERING NATURAL LANGUAGE QUERIES

This invention relates to answering natural language queries.

5

Such a query may be a question phrased in English, for example, and the response may be sentences of text that belong to a body of free-text sources and are responsive to the question.

10 One way to find the relevant sentences of text uses an index that is created in advance. In a simple example, an index could include the words "Georgia" and "capital" and associated pointers to sentences that include those words. At run time, if a question asks about the capital of Georgia, the index can be used to find
15 responsive sentences.

In the invention, implicit references (also known as anaphora in linguistic literature) are inferred from the words of segments of text. In response to a query, one or more segments are identified as
20 relevant to the query based at least in part on the implicit references. Using implicit references improves the quality of the responses to the query.

A characteristic of natural language text is the use of words
25 (references) that refer to other words or to concepts that appear in or are implied by other parts of the text (antecedents). For example, in the sentence "*He is best known for his theory of relativity*," the word "he" (the reference) may refer to the name "Albert Einstein" (the antecedent) that appears in another sentence:
30 "*Albert Einstein was one of the greatest scientists of all time*."

Two broad categorizations of references may be useful. One broad categorization is based on the positions of the antecedent and the reference. The other broad categorization is based on the type of reference. The first categorization is based on three distinct contexts in which the reference may be used in a question answering setting.

References of the kind that are based on position may occur in at least three different contexts in a question answering setting:

1. Between two sentences

S1: *Albert Einstein* was one of the greatest scientists of all time.

S2: *He* is best known for his theory of relativity.

In sentence S2, the word "he" refers to "Albert Einstein" in sentence S1.

20

2. Between two questions

Q1: When was *Einstein* born?

Q2: Did *he* invent relativity?

25

In the question Q2, the word "he" refers to "Einstein" as used in question Q1.

3. Between a question and a sentence

S3: *Einstein* is best known for his theory of relativity.

Q3: *Who* invented relativity?

5 The word "who" in question Q3 refers to "Einstein" as used in sentence S3.

All three types of references may have to be resolved to match a question with responsive sentences in the free-text sources.

10 Consider the following example:

S4: *China* is a huge country in eastern Asia.

S5: *It* produces more cotton, **rice**, and wheat than any other country.

15 Q4: What is the scientific classification of **rice**?

Q5: *Which countries* produce **this crop**?

The phrase "this crop" in Q5 refers to "rice" in Q4. The word "it" in S5 refers to "China" in S4. The phrase "which countries" in Q5 refers to "it" in S5 and in turn to "China" in S4. A resolution of the three types of references would show that S5 is a potential answer to Q5.

25 The second categorization is based on the type of phrase used for the reference and includes the following five groups (examples included):

Pronoun: China is a big country. **It** is in Asia.

Definite Noun Phrase: China is in Asia. This country produces rice.

Name variant: International Business Machines versus IBM, Great Britain versus Britain versus England.

5 **Indirect references:** (in an article about China): The climate is usually mild: (Here the climate does not refer to China but it is known that it is the Chinese climate that is under discussion.)

Indirect references rely on "has-a" relationships.)

Null references: "Cisco acquired Cerent Corp. for 7.5 billion

10 dollars. The negotiations lasted 3.5 months." The second sentence is responsive to the question "How long did Cisco negotiate with Cerent?" even though it does not contain any words that refer to Cisco or Cerent.

15 Implementations of the invention take advantage of references to identify sentences in free-text sources that may answer natural language questions.

20 One goal of some implementations of the invention is to shorten the processing delay in receiving an answer after a question is posed at run time. In general, shifting processing steps from run time to a preliminary indexing phase can reduce the delay.

25 One way to shift processing to the indexing phase relates to the need to match synonyms that appear in a question and in a sentence. For example, the words "produces" and "raise" in the following question and sentence must be matched at run time:

S6: China *produces* more corn than any other country.

Q6: In which countries do people *raise* corn?

By generating and storing synonyms for the word "produces"

during the indexing phase, rather than generating synonyms for

5 "raise" at run time, the processing delay in responding to questions can be reduced, an advantage which justifies the additional storage space required for the larger index.

Another opportunity for shifting processing to the indexing phase

10 relates to the fact that there tend to be many more specializations of a concept than generalizations of a concept. For example, there are more than 250 countries (including China) that represent specializations of the concept "country" but relatively few generalizations for the concept "China". So, in the following 15 example, overall processing time is saved by generating and storing the generalizations of "China", the concept that appears in the sentence, during the indexing phase, rather than generating the larger number of specializations of "countries", the concept that appears in the question:

20

S7: *China* produces more corn than any other country.

S8. In which *countries* do people raise corn?

Thus, in general, in one aspect, the invention features receiving

25 segments of text (e.g., sentences), each segment having elements. Implicit references are inferred from the elements of the segments. A query is received, and, in response to the query, one or more segments are identified as relevant to the query (e.g., by scoring) based at least in part on the implicit references.

Implementations of the invention may include one or more of the following features. The implicit references may be inferred prior to the time when the query is received and may be stored as entries in a searchable index, each entry including a pointer to one of the segments from which the reference was inferred. One or more of the identified segments may be selected for presentation to a user.

5 The implicit references may be generalizations of the elements contained in the segments. The references may be name variations that refer to elements, or indirect references to elements, or definite noun phrase references to elements, or pronouns, or null references. The antecedents of the indirect references may be found in titles or in headings. The antecedent can be a concept 10 15 recognized by a pattern of characters (e.g., a date) and it can be referred to by a generalization (e.g., "when" or "at that time").

20 The scoring may be based on a matching of elements in a question with elements in an index file that contains information about the inferred implicit references. The selection of segments to be displayed may be based on scoring. As few as one segment from a given source need be displayed. The step of responding to the query may include identifying implicit references between the query and a previous query.

25 In general, in another aspect, the features of the invention include receiving a question in the form of natural language speech from a source, automatically recognizing the speech, feeding the recognized speech to a natural language query engine operating on

information accessible through a web site to generate a text answer to the question, synthesizing a spoken response to the question based on the answer, and playing the spoken response back to the source of the question.

5

Implementations of the invention may include one or more of the following features. Commands, also, may be received in the form of natural language speech from a source, the commands may be determined using natural language processing, and the speech may 10 be acted upon by controlling navigation in the web site. The natural language query engine operates by receiving segments of text, each segment having elements, inferring implicit references from the elements of the segments, receiving a query, and, in response to the query, identifying one or more segments as 15 relevant to the query based at least in part on the implicit references.

In general, in another aspect, the invention features speaking a natural language question to a web site and receiving a natural 20 language spoken answer to the question back from the website.

In general, in another aspect, features of the invention include receiving a natural language question from a user, deriving information about the user from the question, selecting 25 promotional information based on the information about the user, generating an answer to the question using a natural language query engine, and returning the answer to the user together with the promotional information.

Implementations of the invention may include one or more of the following features. The information about the user may include preferences suggested by the question. The promotional information may include advertising. Advertising tags may be 5 generated for use in selecting the promotional information. The natural language query engine operates by receiving segments of text, each segment having elements, inferring implicit references from the elements of the segments, receiving a query, and in response to the query, identifying one or more segments as 10 relevant to the query based at least in part on the implicit references.

In general, in another aspect, the invention features receiving page information contained in a web page that is being viewed by a 15 user, deriving user information about the user from the page information using a natural language query engine, selecting promotional information based on the user information, and displaying the promotional information to the user while the user is viewing the web page.

20

In general, in another aspect, the invention features receiving a natural language question from a user, deriving information about the user from the question, selecting available information that is related to the question, generating an answer to the question using 25 a natural language query engine, and returning the answer to the user together with the available information.

Implementations of the invention may include one or more of the following features. The information about the user may include

preferences suggested by the question. The information related to the question may include articles. Advertising tags may be generated for use in selecting the information. The natural language query engine operates by receiving segments of text, each 5 segment having elements, inferring implicit references from the elements of the segments, receiving a query, and in response to the query, identifying one or more segments as relevant to the query based at least in part on the implicit references.

10 In general, in one aspect, the invention features entering a natural language question on a wireless personal electronic device, generating natural language answer to the question using a natural language query engine, and presenting the natural language answer to a user.

15 Implementations of the invention may include one or more of the following features. The question may be entered through a keyboard. The answer may be presented through an interface of the device. The natural language query engine operates by receiving 20 segments of text, each segment having elements, inferring implicit references from the elements of the segments, receiving a query, and in response to the query, identifying one or more segments as relevant to the query based at least in part on the implicit references.

25 In general, in one aspect, the invention features presenting to a user a web page that comprises a shopping cart, displaying on the shopping cart an identification of an item for purchase, providing a mechanism that enables the user, without leaving the shopping

cart, to ask a natural language question, and providing an answer to the natural language question.

5 Implementations of the invention may include one or more of the following features. The mechanism may include a dialog box displayed over the shopping cart web page. The dialog box may be displayed in association with the identification of the item for purchase. The answer may include information about the item for purchase. The user may take a step in response to the answer and

10 complete a transaction on the shopping cart web page. The answer may be provided from a natural language query engine. The mechanism may be provided by an agent that watches items being added to the shopping cart.

15 In general, in another aspect the invention features receiving natural language questions about products, selecting product information using a natural language query engine based on the questions, and serving the product information from a web server to a user.

20 Implementations of the invention may include one or more of the following features. The user may respond to the web server by buying one of the products. The questions may identify desired characteristics of the products.

25 In the invention, the natural language search is done by entering the search in a field of an email message and sending it to an email address.

In general, in another aspect, the invention features a method that includes (a) receiving from a user, over an electronic network, an electronic mail message containing a written natural language query, (b) identifying the written natural language query in the electronic mail message, (c) using a natural language query engine to apply the natural language query to a body of information, to generate information responsive to the query, and (d) taking an action based on the responsive information.

5

10 Implementations of the invention may include one or more of the following features: Taking an action may include sending an electronic mail message containing the responsive information to the user over the publicly accessible electronic network, or filling an order for a product or service. The query may include a question

15 to be answered and the responsive information may include an answer to the question. The query may include a request for an action or service and taking an action may include providing the action or service in response to the request. The body of information may include textual content or commercial

20 information. The natural language query may be identified based on an indicator arranged by the user. The indicator may include a position of the query within the electronic mail message, e.g., within a subject field of the electronic mail message. The electronic mail message may be directed to an address that is

25 prearranged to automatically receive and respond to the natural language query.

In general, in another aspect, the invention features apparatus that includes (a) an electronic mail message server connected to receive

electronic mail messages containing natural language queries from an electronic network and to send electronic mail messages containing responses to the natural language queries to the electronic network, (b) software adapted to identify written natural 5 language queries in electronic mail messages received at the server and to provide information responsive to the natural language queries as electronic mail messages to the server for delivery, and (c) a natural language query engine connected to receive the natural language queries from the electronic mail message server 10 and to apply them to a body of information to obtain the responsive information.

In general, in another aspect, the invention features (a) automatically stripping natural language queries from electronic 15 mail messages, (b) automatically applying the queries to a natural language search engine to generate responsive information, and (c) automatically taking action based on the responsive information. Other advantages and features will become apparent from the following description and from the claims.

20 Some implementations of the invention are illustrated in the block diagrams of figures 1 through 15 and described below.

In some implementations of the invention, free-text sources are 25 prepared for use in answering questions by first applying a preprocessing routine 30, shown in figure 1. First, the text is parsed (32) to identify sentence boundaries. For purposes of parsing, the sentence boundaries are identified using patterns that are manually created, although other approaches could be used. In the manual

approach, patterns are described that identify potential end-of-sentence markers (period, question mark, exclamation point, paragraph break, title break, sometimes quotes, etc.). Then certain alternative uses are eliminated. For example, in the case of a

5 period, the eliminated alternative includes periods that appear at the end of abbreviations and in acronyms and floating point numbers, for example.

Each sentence is marked (34) with a single new line in one

10 implementation, or using markup tags in another implementation. A unique sentence number is assigned (36) to each sentence. The numbers are unique within a single index file. Therefore all sentences (whether or not from different documents) that go into a single index get unique numbers. In another implementation, part

15 of the unique numbers (e.g., the first six digits) are used to encode the article the sentence is coming from and another part (the last four digits) is used to identify the sentence number within the article.

20 Titles and other headings are identified (38) in a manner that depends on the text format. Some formats (like HTML) use markup elements that identify the titles. Plain text sources require pattern-based analysis. Titles also are marked (40) to identify some possible indirect references. An example would be the

25 sentence “The economy is booming.” found in an article entitled “China”. Notice that unlike in the case of the sentence “This *country* produces rice”, none of the words in the sentence “The economy is booming” directly refers to China. However, from the title one can infer that the subject is the Chinese economy. One

way to index the title information is with respect to every sentence in its scope. Another more complicated way to use the title information is to build and make use of a knowledge base of part-whole, group-member relationships. Such relationships would

5 include, for example, the fact that a typical country has a population, an economy, a president, and an army, etc.. Then, when any of these words (e.g., economy and president) are used by itself in a sentence, the indirect reference to the country can be identified. The output of the pre-processing is a pre-processed text

10 file 42. In one implementation, the pre-processed text file has text of one sentence on each line preceded by a sentence number and a tab character and followed by the text of the applicable titles. In another implementation, a special markup language (similar to HTML or XML) may use specific tags to mark sentences,

15 paragraphs, sections, documents and titles in the text. The sentence tags contain id numbers as part of the tag such as: <s id=124345>. This format is more flexible and may be easily extend to include other tags. A user may be permitted to specify references not identified by the indexer by explicitly inserting them

20 into the pre-processed text file using specific tags.

In one implementation, all the text sources that go into a single application (e.g., a whole encyclopedia) can be converted into one large pre-processed text file before being passed to the indexer.

25 Another implementation could use separate pre-processed files for each article and let the indexer read the information from multiple files.

As shown in figure 2, after pre-processing, the indexing phase 50 begins. The purpose of the indexing phase is to use the pre-processed text file 42 to build an index file (table) 70 that lists foreseen ways in which a question may refer to an element of a sentence. A single index file is built for all sources in the system.

By "element" of a sentence, we mean a concept referred to in the sentence. The concept may be referred to using an ordinary word (walk, cake), a name (Bill Clinton), a multi-word phrase (stand up, put on), a pronoun (he referring to Bill Clinton), a definite noun phrase (the country referring to China), an indirect reference (the economy, indirectly referring to China), or a null reference (there is no word referring to the concept but the concept is still referenced). For example, if the text contained the sentences: "The war started in 1939. Germans invaded Poland.", the answer to the question "When did Germans invade Poland?" would be 1939 even though there is no word in the second sentence directly or indirectly referring to this time phrase. Time phrases and place phrases often affect more than a single sentence, therefore creating null-references.)

Each entry in the index file 70 includes a pointer to the sentence to which the questions may refer based on that entry. Conceptually the index file relates the elements found in a sentence to a unique identifier for that sentence. The index file can be thought of as a two-column table in which one column contains sentence ID numbers and the other column contains the words, concepts, referents, generalizations, and synonyms (collectively referred to as the elements of the sentence).

For efficient scoring later, the following three components are created for the index: the string buffer, the sentence id buffer, and the hash table.

5

The string buffer contains the null terminated strings of each element found in the source text. The strings are placed in the buffer consecutively in no particular order.

- 10 The sentence id buffer contains sentence ID arrays for each element. The array for a particular element can be identified by giving the start position in the buffer and length of the array. The arrays are placed in the buffer consecutively in no particular order.
- 15 The hash table is a standard hash table that contains key-value pairs and that enables a fast search of a given key. The key of each entry is a pointer to the string buffer. The value of each entry consists of a pointer to the sentence ID buffer and an array length.
- 20 This structure enables finding the sentences that contain a particular element as follows: First, the element is searched in the hash table by comparing it with certain keys in the hash table. For each comparison, the string in the string buffer that the key points to is retrieved and compared to the element. When a match is
- 25 found, the corresponding sentence ID buffer pointer and array length is read. Finally, the specified array is located in the sentence ID buffer.

In the indexing phase, each sentence in the preprocessed text file 42 is read and passed to several modules. Each module reads the words of a sentence and, based on them, recognizes certain types of constructions and references that represent foreseen ways in 5 which a question may refer to an element of the sentence. When a module identifies one of those ways, it writes an entry into the index file 70 together with the unique identifying number of the sentence from which it was generated.

10 In one implementation, there are eight indexing modules called: **words, title, word-isa, ako, patterns, names, name-isa, and references.**

As shown in figure 2, the **words** module identifies (50) each word 15 in the current sentence and adds it to the index file. The **words** module also derives the stem of each word. using a table of English word and word stem pairs, such as . flowers->flower and went->go. The **words** module adds the stem to the index file for use, for example, in matching morphological variants of words that 20 may appear in a question.

In the **title** module, the words in each heading in the set of headings that apply to a sentence are added (52) to the index file with pointers to the sentence. In one implementation only one 25 heading (the document title) is used for every sentence in a document. In another implementation, the pre-processed text file contains tags for titles of various levels (document, chapter, section, subsection, for example) and sectioning tags that identify the scope of each title. Using these tags, the indexer is able to

determine, for each sentence, the document, the chapter, and the section that it is in. The indexer combines all titles that apply and indexes them with the sentence. Title indexing may not be appropriate for every source. For example, encyclopedia sources 5 have well defined titles that are usually appropriate and helpful whereas newspapers have partial sentences for titles, which are usually not appropriate for the above method.

The **word-isa** module generates (54) the generalizations 10 (mentioned earlier) for words that appear in the sentence and for words that appear in headings. For example, if the word "red" appears in a sentence, the generalization word "color" is placed in the index file so that a question that asks "what color" will be matched to the sentence that includes "red". For this purpose, a 15 database table with the same name (**word-isa**) and containing two columns is used. The first column contains words and the second column contains possible generalizations. For example, "red->color" would be one of the entries in that table.

20 The **ako** module identifies generalizations (56) of generalizations already generated. For example, if the **ako** module encounters the generalization "color" that had been generated at step 54, the **ako** module adds the further generalization "attribute" to the index file.

25 The **patterns** module reviews (58) the text for special patterns of dates and numbers and adds the generalizations to the index file. For example, if the date January 23rd, 1998, appears in the text, the patterns module would add the generalizations "date" "time" and "when" so that when a question asks "when did this event

happen?” it matches the date. Another example that appears frequently in an encyclopedia is the lifespan information in biographies. The first sentence of a typical biography starts “John Doe (1932-1987) … ”. A pattern that recognizes the life-span structure allows matching of questions of the type “When was John Doe born?”

5 The **names** module identifies proper names (60) in the text and generates and indexes the names accordingly. For example, the **names** module uses two methods to identify names in a sentence. The first method uses a list of precompiled names and name variations to match those in the sentence. For example “United States” and its variations “U.S.A.” and “United States of America” would be in the name list and each would be recognized as a name

10 15 The second method uses patterns that identify names and name types. Proper names are marked with capitalization and can be isolated easily. (There are some difficulties associated with sentence beginnings and small function words like “of” that are not capitalized in the middle of a name.)

20 The **names-isa** module generates generalizations (62) for proper names and adds them to the index file. For example, if the name “Clinton” is found in the text, the word “President” could be added to the index file. Other examples are “China -> country” and

25 “Albert Einstein -> physicist”. The name generalization makes use of a knowledge based and a pattern based method as well. If a name is found in the database, generalizations of the name are located in the **name-isa** table. This is a table just like the **word-isa** table that lists one or more generalizations for a given name. For

names that were not found in the table but that were detected using capitalization, for example, the rough generalization of the name (person, place, organization) can be inferred using internal and external clues. An example of an internal clue would be the 5 appearance of the word "Corp." as part of the name, which would imply that it is a company. Similarly "Mount" or "City" implies a place and "Mr." or "John" implies a person. External clues are words outside the name that provide information. For example, if a name is preceded by "in" one can deduce that it is a place or 10 possibly an organization but not a person.

The **references** module identifies (64) implicit references in the form of pronouns, definite noun phrases and name variants. The module could also handle indirect references and null references. 15 (Handling indirect references would require a "has-a" table similar to the "is-a" table discussed below. The "has-a" table would represent relationships of the kind: "A country has an economy, a president, an army, etc."

20 Antecedents of references are determined using a short-term buffer 80. The antecedents are added to the index file, and the short term buffer 80 is updated with the potential references for the new names in the sentence, in the following way:

25 The short-term buffer contains a set of pairs of the type "he -> Bill Clinton", "country -> China", i.e. a potential reference pointing to a potential antecedent. The sentence is scanned for potential reference words or phrases. For each one discovered, the set of the potential antecedents is added to the index file. After each

sentence is processed, the short-term buffer is cleared and updated with new potential references. The new potential antecedents are the names and other concepts used in the current sentence (either explicitly mentioned or implicitly referred to). The new potential 5 references are all generalizations, name variants and pronouns compatible with these antecedents.

The short-term buffer 80 has two fields. One field contains antecedent words, the other contains potential references 10 associated with each of the antecedents. As each element of a sentence is encountered, potential references are stored in the short-term buffer (e.g., when "China" is encountered in a sentence, the potential references "country", "nation", and "it" are added to the potential references field). When a referring word or phrase 15 such as a pronoun or a definite noun phrase (e.g., "the country") is encountered in a later portion of the text, the word is looked up in the short-term buffer to identify the possible antecedents.

The modules that are active during the indexing phase use the 20 following lexical databases to perform their functions.

A **skip-word** database 82 lists function words such as prepositions, conjunctions, and auxiliary words that are not to be added to the index file. The **skip-word** database is used in step 50 of figure 2.

25 A **stem** database 84, also used in step 50, contains a list of the stems of most English words. The word stems can be found in sources such as the CELEX lexical database available from the Linguistic Data Consortium of the University of Pennsylvania.

Other sources for this material include on-line dictionaries.

Alternatively, one could use a rules-based approach by analyzing a word and stripping its suffixes.

5 A **word-isa** database 86, used in step 54, contains generalizations of single words that can potentially match question words. The **word-isa** table is generated using three approaches: 1. Consulting online lexical (“word-related”) databases like wordnet or thesauri like Roget's. 2. Writing data-mining programs that process large
10 corpora (text sources) or the actual source to be indexed as a way to discover such relations. 3. Manually editing and cleaning up the results of 1 and 2. A source like an encyclopedia typically includes an article classification and a title index which contain useful information related to the generation of the **isa** and **ako**
15 tables.

An **ako** database 88 contains lists of generalizations for single words and is used in step 56. The **ako** database is generated in a manner similar to the generation of the **word isa** table.

20 A **name-isa** database 90 contains generalizations for recognized proper names like countries, companies, and famous people and is used in step 62. The **name isa** database is generated in a manner similar to the generation of the **word isa** table. The pattern-based
25 rules mentioned before (which assign person/place/organization type general classes to names) can be used to expedite the process.

After the indexing phase, scores are generated (92) for each unique sentence element contained in the index file. The score is inversely

proportional to the number of times the sentence element appears in the index file.

The score also reflects the part of speech and the confidence in reference resolution. The score is stored in a score file 94.

5

In one implementation of the scoring algorithm, the score file contains a set of pairs of the type, for example, "walk -> 7.86", "Clinton -> 15.76". The numbers are computed based on the frequency of the given term, e.g., as $-\log_2$ (frequency). The frequency is either computed based on the index file by counting the number of occurrences of each term in the index file or based on a large reference corpus (such as the Cob corpus frequencies from CELEX). The latter is particularly useful when the data to be indexed is small and its frequencies are not statistically significant.

10 The score file may then be manually modified to assign higher values to domain-specific terms or lower values to optional modifiers.

15

The index file is in the form of a set of pairs of the type "walk -> 132459", "Clinton -> 345512" etc. The numbers are unique sentence ID numbers. Here is an example sentence and some sample terms that are inserted into the index file for this sentence:

Sentence: He was the one of the brothers of the apostle Peter.

Example terms:

25 **Plain word:** apostle

Stem: brother (from brothers)

Generalization: person (from apostle via **word-isa** file)

Indirect reference: Andrew ("he" refers to Saint Andrew in the previous sentence).

Once the indexing phase is completed, the index file and score file can be used as the basis for answering questions.

5 As shown in figure 3, the **run time** process (100) receives questions posed by a user and uses the index file and the score file to identify sentences that may answer the questions. The run time process has two main parts. One part is the analysis of the questions 101 to produce a question file 104. The second part is the 10 matching of information 103 in the question file with information in the index file to identify sentences that are likely to provide answers to the questions.

In the first part of the run time process, each word in a question is 15 processed using modules similar to those used in the indexing phase.

A **stems** module 102 uses the skip-word database 82 to pass over certain words and uses the stem database 84 to determine stems of 20 each word and records them in the question file 104.

A **q-ref** module identifies (106) potential references between the current question and antecedent elements of other questions. The identification is done in a manner similar to step 64 in figure 2, 25 using a short-term buffer 105. The antecedents are recorded in the question file 104.

No generalizations, synonym generation, etc. are performed at run time. It is important that such steps not be performed at run time to avoid double matching.

- 5 The matching part of the run time process searches in the index file for each element in the question file 108. If an element in the question file is found in the index file, an answer score for the sentences associated with that element is updated by adding the score 108 associated with that element in the score file 94.

10

After all elements in the question have been matched, the sentences are sorted 112 according to their respective total scores.

- 15 Using the sorted sentence list, a decision 114 is made about which sentences to display as the answer to the question.

One approach is to display sentences that are at the top of the scoring. By comparing the sentences having the highest scores with the maximum possible sentence score, a determination can be

- 20 about the quality of the answer represented by each of those sentences. A typical noun in English is worth about 10 to 15 points. A sentence that has a score within 10 points of the maximum possible score would represent a high quality answer. If the answer quality of the highest scoring sentence is high, that
- 25 sentence could be displayed alone. If several of the top-scoring sentences have close scores, they can all be displayed. A bias can be applied to cause the display of high-scoring sentences from different free-text sources in lieu of multiple sentences from a single source.

If the highest scoring sentence is not a high quality answer, or if the question is a "how" or "why" question, additional context around the sentence can be displayed to aid the user's interpretation.

5 For this purpose, the display algorithm can be configured to display one or two neighbor sentences around the sentence or the whole paragraph around the sentence.

If the highest scoring sentence is a low quality answer, the user 10 could be told that no good answer was found and a few pointers to relevant documents could be displayed.

The answer system is useful in a wide variety of contexts, including the Internet, local networks, or a single workstation. In 15 the case of the Internet, the indexing can be done at a central location and the run time process can handle questions received from browsers at a central server.

The invention offers a number of advantages. In particular, the 20 quality of the answers is high because the indexing of implicit references significantly improves the chances that useful responsive sentences will be found. The invention is useful in a wide variety of contexts, among them on-line searching using the World Wide Web.

25

Other implementations are within the scope of the claims.

For example, portions of text other than sentences, such as paragraphs or sections or chapters can form the basis of the

indexing and scoring. Also, other kinds of references and generalizations could be used as the basis for the indexing phase.

Questions need not be phrased as complete English sentences.

5

Languages other than English can be used.

Indexing need not be captured in a single central index file and score file but can be distributed among multiple index files and
10 score files. At run time, questions may be answered by a scoring system that operates on all of the files.

Other types of references (null, indirect) can easily be integrated into the existing framework once the necessary knowledge is built.

15 Also, once grammatical relations are determined with satisfactory accuracy, they can be incorporated into the existing indexing-retrieval framework without major changes to the architecture.

A variety of other applications may make use of the query
20 response techniques discussed above. Among the applications are the following:

1. As shown in figure 4, a person could use any voice-based communication device, such as a wireless or wired phone, to
25 connect (200) to a web site, and using voice, navigate the web site and obtain information by issuing voice commands and questions. The user could utter a natural language query (202). The website would include speech recognition software that would permit voice-to-text transcription of the query (204). The text would then

be passed to the query response engine described earlier (206). The query response engine generates one or more responses (208) in text form and passes them to a speech synthesizer (210). The speech synthesizer converts the text to speech (212) that is played 5 back over the phone to the user (214).

2. As shown in figure 5, a person could get answers to questions from a wireless communication device. After the device is connected to a web site 220. the user types a query on the 10 device, either using a keyboard or a stylus only a touch-sensitive screen. At the website, the query is passed to the query response engine described earlier (224). The query response engine generates responses (226) that are in the form of answers to the query rather than in the form of links to places where the answer 15 may be available. The answers are then returned to the wireless device (228). For example, the question entered by the user might be "What was one of Einstein's achievements?" One response might be the answer "Einstein developed the theory of relativity."

20 3. As shown in figure 6, advertising delivered to a web user can be personalized based on questions that the user asks. The user enters a query (230). As before, the text of the query is passed to the engine (232) and a response is generated (234). The engine also uses the response to generate ad TAGS (238). For example, if the 25 question is "what are the ski conditions like in Aspen?" the engine will generate TAGS that relate to commerce for Aspen, such as "Ski Rental, Cabin Rental, Dining in Aspen, Flying to Aspen". These TAGS are then used to extract appropriate ads from ad

inventory. The ads are presented to the user along with the answer to the question asked.

4. As shown in figure 7, a user browses the web (250). Based
5 on a web page being displayed to the user in the course of the
browsing, a set of information, for example, words that appear on
the web page, is derived for use with the query response engine
(252). The information is applied to the query response engine as if
it were a query (254). The results of the query are used to generate
10 ad TAGS (256) and the TAGS are used to extract appropriate ads
from ad inventory (258) as before. The ads are presented to the
user as part of the page being read, or a later page (260).

5. As shown in figure 8, in an application similar to the one
15 described in figure 6, except that the TAGS are chosen 270 to
relate to articles or information, for example, about Aspen, such as
"latest Aspen news, Traveling in Aspen, Events in Aspen", etc.
These TAGS are then used to extract appropriate information from
information sources (280) and construct the next page that is
20 shown to the user (282). The resulting personalized page is then
presented to the user along with the answer to the question asked
(284).

6. As shown in figure 9, another application develops user
25 profile and preference information based on questions asked. A
user types (or asks) questions 290. The query response engine
processes the questions (292) and generates a log (294) that
includes the following information, for example: identity of the
user (name, IP address, etc.); the questions asked and answers to

the questions; any un-answered questions; and the click stream reflecting what the user did after the answers were delivered to him. The log is analyzed (296) to generate profile TAGs. The profile TAGS are used to update a user profile (298). The next 5 time the user logs in, or enters another query, the updated profile is used to personalize web pages and advertising for the user (300).

7. As shown in figure 10, another application facilitates on-line shopping by answering questions about products in the 10 shopping cart. The user adds items to a shopping cart on a commercial web site (310). The items are used as the basis for generating question dialog boxes for each of the items (312). Each dialog box hovers above the shopping cart. The user may then ask a question about an item (314). The query response engine answers 15 the question without forcing the user to leave the shopping cart (316). The answer is shown in the hovering dialog box. The user completes the transaction based on the answer (318).

8. As shown in figure 11, in another application, the user can 20 navigate, e.g., a product catalogue by asking questions. The user (in plain language) asks the system to show products that meet specific criteria (e.g., "show me the cheapest PC", "the fastest car", etc.) (330). The query response engine processes the request (332) and generates a list or an item that meets the criteria (334). The 25 user then clicks on the items to buy (336).

9. As shown in figure 12, in another application, corporate and departmental reports can be generated based on a question log. Users ask questions to interact with a reporting system (338). The

query response engine processes the questions (340) and updates the question log (342). The log includes the following information: identification of the user (name, IP address, etc.); questions asked, and answers to the questions; un-answered questions; and the click stream indicating what the user did after the answer was given. The log is analyzed by a report generator to generate pre-defined reports (344). The reports use question subjects, frequencies, users, whether they were answered or not, and other information contained in the questions to surmise information that is relevant to various departments such as product development, support, finance, human resources, etc. The reports are interpreted by humans to make business decisions about new products, product design, financing, internal processes, control, and other aspects of the business. The reports are based on context and intelligence extracted from the questions the users ask.

Another application, shown in Figures 13 through 15 is useful in an email context.

A shown in figure 13, a natural language query 10 has been typed into a subject field 12 of an electronic mail (email) message 14. The message also includes message field 16 (which is shown empty but could contain other information), a “to” address 17, for example, an Internet address of an email message server, and a “from” address 18, which typically identifies the source of the message.

By a natural language query we mean any arbitrary clause or sentence that is expressed in a human language, such as English, in

a manner that is natural to native users of the language. Among other things, the query need not comply with any special syntax or vocabulary to accommodate to the needs of a computer program. The query need not be expressed as a complete sentence or as a 5 question. It could be expressed, for example, as a command, or an order, or any kind of request for action or service.

As shown in figure 13, the email message containing the query is sent through the Internet 20 to the "to" address which is the 10 location of an email message server 22. The email message server automatically receives the messages, automatically strips out the subject information, in this case the natural language query, and the "from" address, and automatically passes the query to a natural language query engine 24. The natural language query engine 24 15 applies the query to a body of information 25 that may contain a response or responses to the query. The resulting response or responses 28 are passed back through software 26 which forms a new email message 40 (such as the one shown in figure 14) using the response 28 in the message field 42 and the received "from" 20 address as the "to" address 44 of the new message. The new message is forwarded to the email message server, which sends it through the Internet back to the source of the query to which it responds.

25 As mentioned earlier, the queries could be expressed as commands, orders, or any requests for action or service.

For example, a message sent to an email message server 48 of an on-line grocery site 50 could contain an order: "For Customer

#123, please deliver 1 pound of cherries by 5PM tonight". The on-line site can receive this message, use a natural language query engine 52 to apply it to stored commercial information 54, for example, a database of address and credit card information 5. associated with the user, and then apply the information in the database to software 56 that builds an order, charges the customer, arranges for the delivery of the goods, and e-mails back a confirmation of the delivery.

10 Another example is, "Send a message to my staff to attend a 5PM meeting in the office, tomorrow!" which could be handled in a similar way.

15 Another example is "please ship B2305 printer to customer 123 via overnight delivery, and charge it to account 123". The natural language query engine will interpret the instruction, go into a product database and find B2305, place the order in the name of customer 123 for overnight delivery, and charge account 123.

20 Another example is, "sell 100 shares of GE in account 123". The natural language engine will interpret the order, translate it into transactions, and e-mail back a confirmation.

25 Any natural language query engine could be used to respond to the queries. One suitable engine is described above.

For example, the query could be identified by other means than positioning it in the subject field. The query could be written in the message. Within the message field, the query could be

distinguished from other text by predefined markers. For example the query could be preceded and followed by the string ++**. Or the query could be placed on the first line of the message.

5 The response to the query could be returned other than in an email message, for example, by FAX or by posting on a website.

Other instructions could be provided from the source to the email message server with respect to the query. For example, the email
10 message could contain credit card or other charge information and the user could be charged for the response service. Or the message field could contain instructions about how to return the response, for example, including a FAX phone number.

15 The natural language messages can be sent and received over a wired or wireless network or a point-to-point connection. A user could speak the natural language message into a cellular or mobile phone. At the phone, or centrally, the message can be recognized and converted into text to be applied to the natural language query
20 engine.

Other implementations are within the scope of the claims.

Claims

1. A method comprising
receiving segments of text, each segment having elements,
inferring implicit references from the elements of the
segments,
receiving a query, and
in response to the query, identifying one or more segments
as relevant to the query based at least in part on the implicit
references.
- 5 10 2. The method of claim 1 in which the implicit references are
inferred prior to the time when the query is received.
- 15 3. The method of claim 1 in which the implicit references are
stored as entries in a searchable index, each entry including a
pointer to one of the segments from which the reference was
inferred.
- 15 4. The method of claim 1 in which the segments comprise
sentences.
- 5 20 5. The method of claim 1 further comprising
selecting one or more of the identified segments for
presentation to a user.
6. The method of claim 5 in which the segments that are
presented to the user are determined based on scoring.
7. The method of claim 5 in which only one segment is
displayed.
- 25 8. The method of claim 5 in which only a single segment from
a given source is displayed.
9. The method of claim 1 in which the implicit references
comprise generalizations of specializations represented by the
elements contained in the segments.

10. The method of claim 1 in which the implicit reference comprises a name variation that refers to an element.
11. The method of claim 1 in which the implicit reference comprises an indirect reference to an element.
- 5 12. The method of claim 1 in which the implicit reference comprises a pronoun.
13. The method of claim 1 in which the implicit reference comprises a definite noun phrase.
14. The method of claim 1 in which the implicit reference 10 comprises a null reference.
15. The method of claim 1 in which antecedents of the implicit reference are found in a title.
16. The method of claim 1 in which antecedents of the implicit reference are found in a heading.
- 15 17. The method of claim 1 in which the implicit reference comprises a generalization and the element to which the implicit reference refers comprises a specialization.
18. The method of claim 1 in which an antecedent may be a pattern of characters and the pattern is referred to by a 20 generalization.
19. The method of claim 1 in which the implicit reference comprises a proper name and the element to which the reference refers comprises a noun or noun phrase.
20. The method of claim 1 in which the implicit reference 25 comprises a pronoun, definite noun phrase, or name variant.
21. The method of claim 1 in which the identifying comprises scoring.

22. The method of claim 21 in which the scoring is based on a matching of elements in a question with elements in an index file that contains information about the inferred implicit references.
23. The method of claim 1 in which responding to the query includes identifying implicit references between the query and a previous query.
24. A method comprising
 - receiving segments of text, each segment having elements, inferring implicit references from the elements of the
 - 10 segments, and storing an index file based on the implicit references for later use in responding to a query.
25. A method comprising
 - receiving a query, and
 - 15 in response to the query, identifying one or more segments of text as relevant to the query based at least in part on implicit references that were pre-stored in an index file.
26. A method comprising
 - receiving a question in the form of natural language speech
 - 20 from a source, automatically recognizing the speech,
 - feeding the recognized speech to a natural language query engine operating on information accessible through a web site to generate a text answer to the question,
 - 25 synthesizing a spoken response to the question based on the answer, and
 - playing the spoken response back to the source of the question.

27. The method of claim 26 also including
receiving commands in the form of natural language speech
from a source,
automatically recognizing the speech,
5 determining the commands using natural language
processing, and
acting on the speech by controlling navigation in the web
site.
28. The method of claim 26 in which the natural language
query engine operates by
10 receiving segments of text, each segment having elements,
inferring implicit references from the elements of the
segments,
receiving a query, and
15 in response to the query, identifying one or more segments
as relevant to the query based at least in part on the implicit
references.
29. A method comprising
speaking a natural language question to a web site, and
20 receiving a natural language spoken answer to the question
back from the website.
30. A method comprising
receiving a question in the form of natural language from a
source,
25 feeding the question to a natural language query engine
operating on information accessible through a web site to generate
a text answer to the question,
returning the text answer to the source of the question.

31. A method comprising
 - receiving a natural language question from a user,
 - deriving information about the user from the question,
 - selecting promotional information based on the information
- 5 about the user,
 - generating an answer to the question using a natural language query engine, and
 - returning the answer to the user together with the promotional information.
- 10 32. The method of claim 31 in which the information about the user includes preferences suggested by the question.
33. The method of claim 31 in which the promotional information comprises advertising.
34. The method of claim 31 also including generating advertising tags for use in selecting the promotional information.
- 15 35. The method of claim 31 in which the natural language query engine operates by
 - receiving segments of text, each segment having elements,
 - inferring implicit references from the elements of the
- 20 segments,
 - receiving a query, and
 - in response to the query, identifying one or more segments as relevant to the query based at least in part on the implicit references.
- 25 36. A method comprising
 - receiving page information contained in a web page that is being viewed by a user,
 - deriving user information about the user from the page information using a natural language query engine,

selecting promotional information based on the user information,

displaying the promotional information to the user while the user is viewing the web page.

5 37. The method of claim 36 in which the information about the user includes preferences suggested by the web page that is being viewed.

38. The method of claim 36 in which the promotional information comprises advertising.

10 39. The method of claim 36 also including generating advertising tags for use in selecting the promotional information.

40. The method of claim 36 in which the natural language query engine operates by

receiving segments of text, each segment having elements,

15 inferring implicit references from the elements of the segments,

receiving a query, and

in response to the query, identifying one or more segments as relevant to the query based at least in part on the implicit

20 references.

41. A method comprising

receiving a question or command from a user,

deriving information about the user from the question or command,

25 selecting promotional information based on the information about the user,

generating an answer to the question or command using a natural language query engine, and

returning the answer to the user together with the promotional information.

42. A method comprising

receiving a natural language question from a user,

5 deriving information about the user from the question, selecting available information that is related to the

question,

generating an answer to the question using a natural language query engine, and

10 returning the answer to the user together with the available information.

43. The method of claim 42 in which the information about the user includes preferences suggested by the question.

44. The method of claim 42 in which the information related to 15 the question comprises articles.

45. The method of claim 42 also including generating advertising tags for use in selecting the information.

46. The method of claim 42 in which the natural language query engine operates by

20 receiving segments of text, each segment having elements, inferring implicit references from the elements of the segments,

receiving a query, and

25 in response to the query, identifying one or more segments as relevant to the query based at least in part on the implicit references.

47. A method comprising

receiving natural language questions from a user,

using a natural language query engine to provide natural language answers to the questions,

enabling the user to take steps through a user interface after questions are received or answers are provided,

5 generating a log of information about the questions, answers, and steps of the user,

in real-time or in batch mode, updating a user profile based on the log,

using natural language processing to extract meaning from

10 the questions asked, answers provided and actions taken by the user based on each question and answer pair, and

selecting content for web pages that are served to the user based on the user profile.

48. A method comprising

15 receiving natural language questions from a user,

using a natural language query engine to provide natural language answers to the questions,

enabling the user to take steps through a user interface after questions are received or answers are provided,

20 generating a log of information about the questions, answers, and steps of the user,

analyzing the log using natural language processing to generate reports.

49. The method of claim 48 in which the log is analyzed with

25 respect to subjects of the questions, frequencies, time stamps, users, and whether answers were given.

50. The method of claim 48 in which the log is analyzed with respect to subject of the questions, frequencies, users, answers or lack thereof, and reports are summarized by categories specified by

the users. Natural language processing is used to map categories to types of data in the log. For example, if the users requests a summary of all questions (or answers) that relate to “system crashing”, natural language processing identifies all questions (or answers) that contain phrases or words that are synonymous to “system crashing”.

5 51. The method of claim 48 in which the log is analyzed with respect to the meanings that can be extracted from questions, the frequency of questions, question types, time of the questions, and
10 users.

52. A method comprising
receiving a natural language command from a user,
deriving information about the user from the command,
selecting available information that is related to the
15 command,
generating an answer to the command using a natural
language query engine, and
returning the answer to the user together with the available
information.

20 53. A method comprising
receiving natural language commands from a user,
using a natural language query engine to provide natural
language answers to the commands,
enabling the user to take steps through a user interface after
25 commands are received or answers are provided,
generating a log of information about the questions,
answers, and steps of the user,
in real-time or in batch mode, updating a user profile based
on the log,

using natural language processing to extract meaning from the commands, answers provided and actions taken by the user based on each command and answer pair, and

selecting content for web pages that are served to the user

5 based on the user profile.

54. A method comprising

receiving natural language commands from a user,

using a natural language query engine to provide natural language answers to the commands,

10 enabling the user to take steps through a user interface after commands are received or answers are provided,

generating a log of information about the commands, answers, and steps of the user,

analyzing the log using natural language processing to 15 generate reports.

55. A method comprising

entering a natural language question or command on a wireless personal electronic device,

generating a natural language answer to the question or 20 command using a natural language query engine, and

presenting the natural language answer to a user.

56. The method of claim 55 in which the question or command is entered through a keyboard.

57. The method of claim 55 in which the answer is presented 25 through an interface of the device.

58. The method of claim 55 in which the natural language query engine operates by

receiving segments of text, each segment having elements,

inferring implicit references from the elements of the segments,
receiving a query, and
in response to the query, identifying one or more segments
5 as relevant to the query based at least in part on the implicit references.

59. A method comprising
presenting to a user web page that comprises a shopping cart,
10 displaying on the shopping cart an identification of an item for purchase,
providing a mechanism that enables the user, without leaving the shopping cart, to enter a natural language question or command, and
15 providing an answer to the natural language question or command.

60. The method of claim 59 in which the mechanism comprises a dialog box displayed over the shopping cart web page.

61. The method of claim 59 in which the dialog box is
20 displayed in association with the identification of the item for purchase.

62. The method of claim 59 in which the answer comprises information about the item for purchase.

63. The method of claim 59 in which the user takes a step in
25 response to the answer and completes a transaction on the shopping cart web page.

64. The method of claim 59 in which the answer is provided from a natural language query engine.

65. The method of claim 59 in which the mechanism is provided by an agent that watches items being added to the shopping cart.

66. A method comprising

5 receiving natural language questions or commands about products,

selecting product information using a natural language query engine based on the questions or commands, and serving the product information from a web server to a

10 user.

67. The method of claim 66 in which the user responds to the web server by buying one of the products.

68. The method of claim 66 in which the questions identify desired characteristics of the products.

15 69. The method of claim 66 in which the natural language query engine operates by

receiving segments of text, each segment having elements, inferring implicit references from the elements of the segments,

20 receiving a query, and

in response to the query, identifying one or more segments as relevant to the query based at least in part on the implicit references.

70. A method comprising

25 receiving from a user, over an electronic network, an electronic mail message containing a written natural language query,

identifying the written natural language query in the electronic mail message,

using a natural language query engine to apply the natural language query to a body of information, to generate information responsive to the query, and

 taking an action based on the responsive information.

- 5 71. The method of claim 70 in which taking an action includes sending an electronic mail message containing the responsive information to the user over the publicly accessible electronic network.
- 10 72. The method of claim 70 in which the query includes a question to be answered and the responsive information includes an answer to the question.
- 15 73. The method of claim 70 in which the query includes a request for an action or service and taking an action includes providing the action or service in response to the request.
- 20 74. The method of claim 70 in which the body of information includes textual content.
- 25 75. The method of claim 70 in which the body of information includes commercial information.
- 30 76. The method of claim 70 in which the action includes filling an order for a product or service.
- 35 77. The method of claim 70 in which the natural language query is identified based on an indicator arranged by the user.
- 40 78. The method of claim 77 in which the indicator comprises a position of the query within the electronic mail message.
- 45 79. The method of claim 78 in which the position is within a subject field of the electronic mail message.

80. The method of claim 70 in which the electronic mail message is directed to an address that is prearranged to automatically receive and respond to the natural language query.

81. A method comprising

5 receiving from a user, over a publicly accessible electronic network, an electronic mail message containing a written natural language query in a subject field of the message, the message being received at an address that is prearranged to automatically receive and respond to the natural language query,

10 automatically obtaining the natural language query from the subject field,

using a natural language query engine to apply the natural language query to a body of information, to generate information responsive to the query, and

15 taking an action based on the responsive information.

82. Apparatus comprising

an electronic mail message server connected to receive electronic mail messages containing natural language queries from an electronic network and to send electronic mail messages 20 containing responses to the natural language queries to the electronic network,

software adapted to identify written natural language queries in electronic mail messages received at the server and to provide information responsive to the natural language queries as 25 electronic mail messages to the server for delivery, and

a natural language query engine connected to receive the natural language queries from the electronic mail message server and to apply them to a body of information to obtain the responsive information.

5 83. A method comprising
 automatically stripping natural language queries from
 electronic mail messages, and

 automatically applying the queries to a natural language
 search engine to generate responsive information, and

10 automatically taking action based on the responsive
 information.

84. The method of claim 70 in which the written natural
 language query is derived by recognition of a spoken natural
 language query.

15 85. A method comprising

 receiving from a user, over an electronic network, a spoken
 electronic mail message containing a natural language query,
 identifying the natural language query in the spoken
 electronic mail message,

20 using a natural language query engine to apply the natural
 language query to a body of information, to generate information
 responsive to the query, and

 taking an action based on the responsive information.

1/8

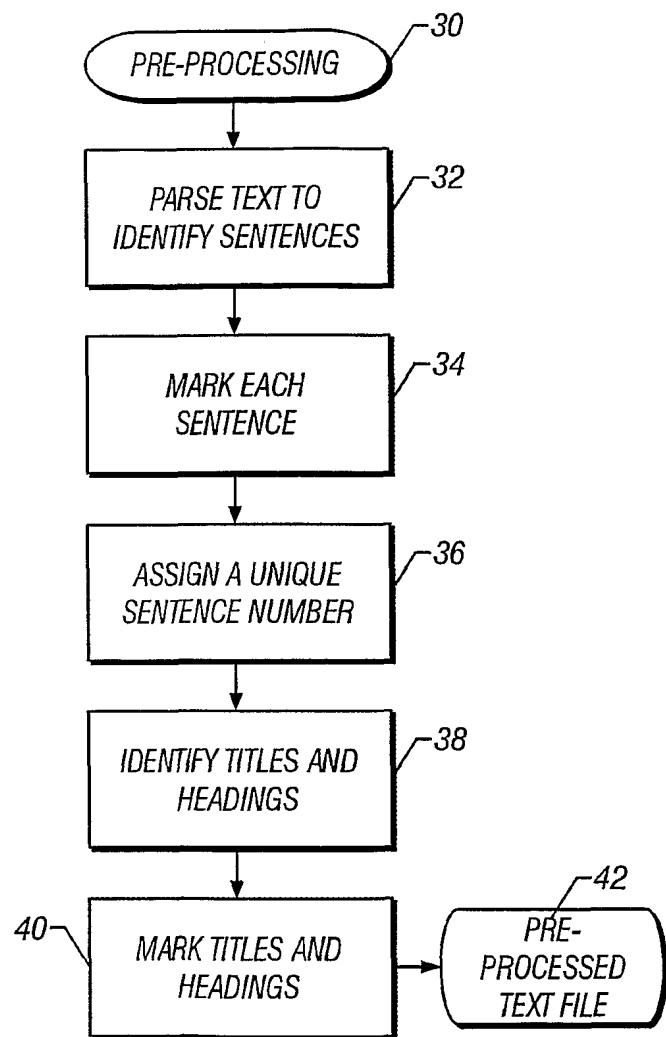
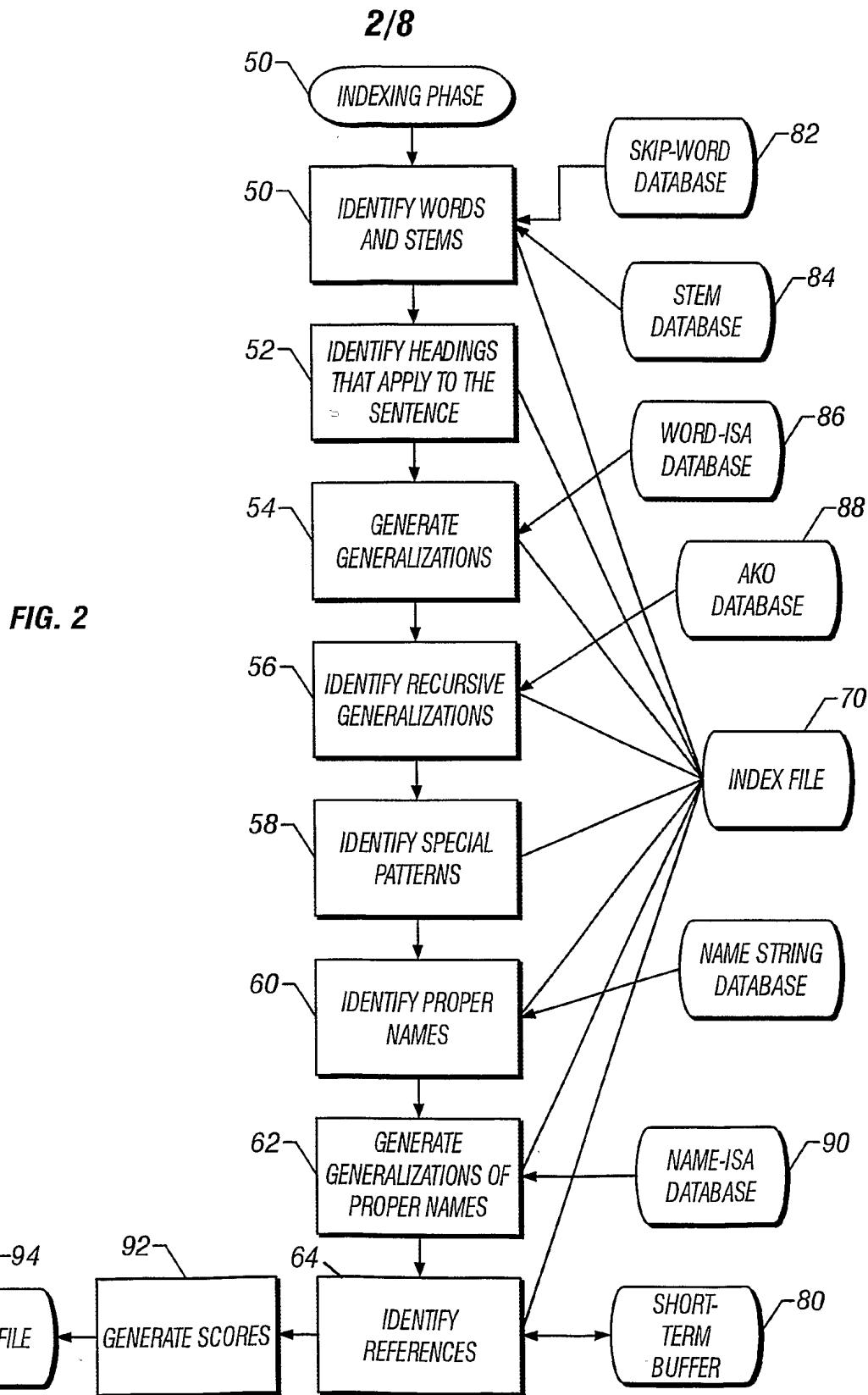


FIG. 1



3/8

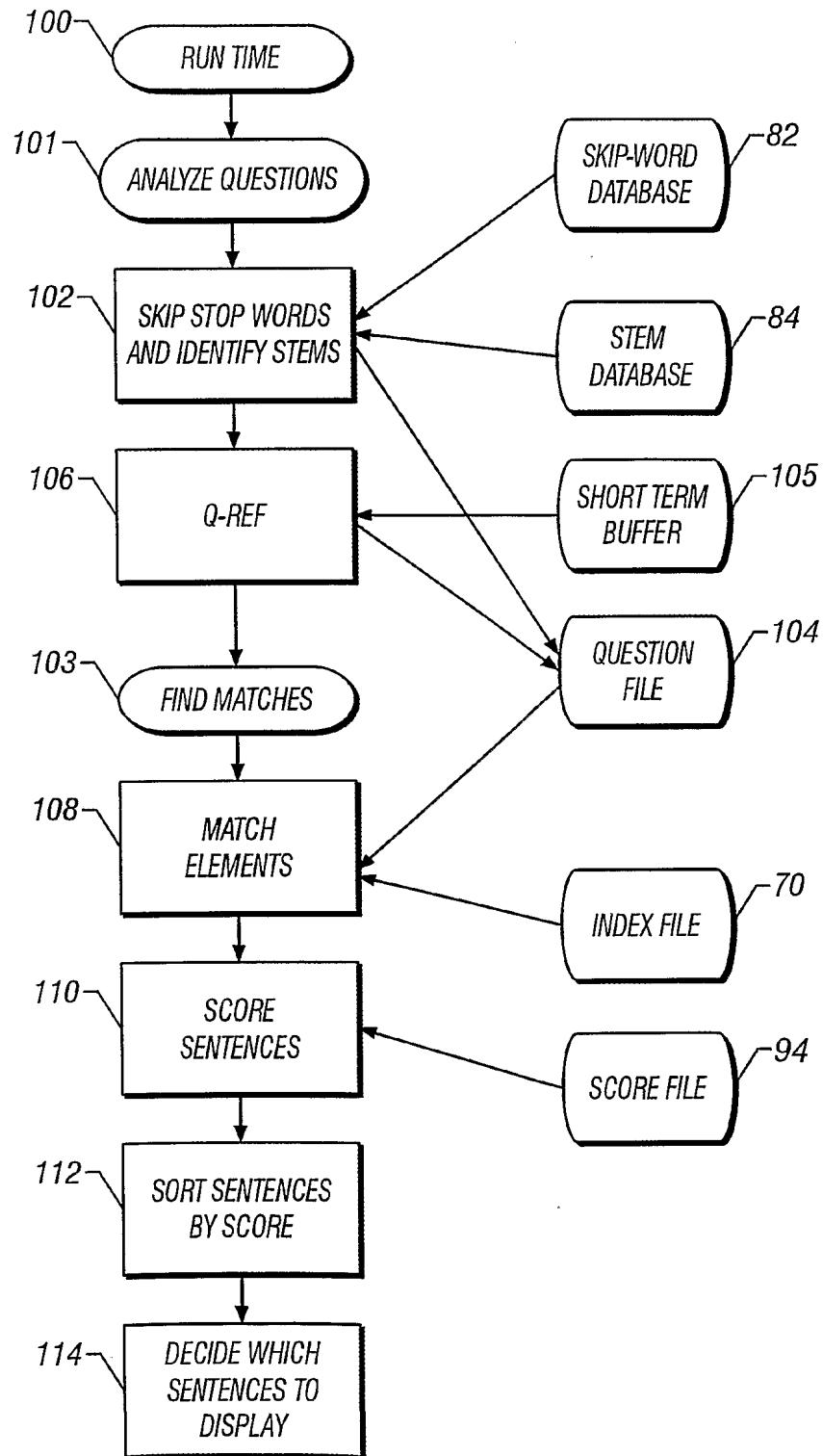
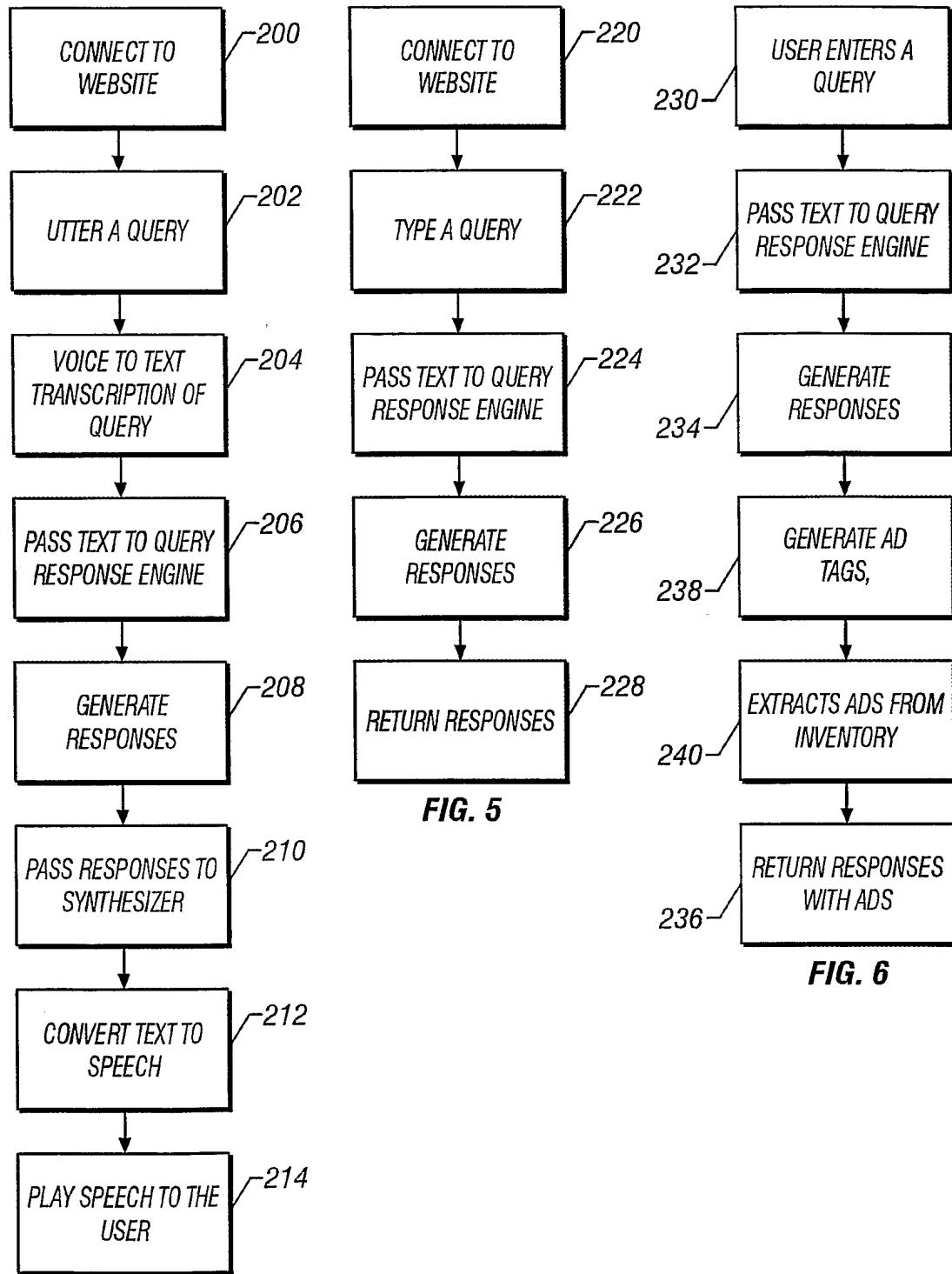


FIG. 3

4/8



5/8

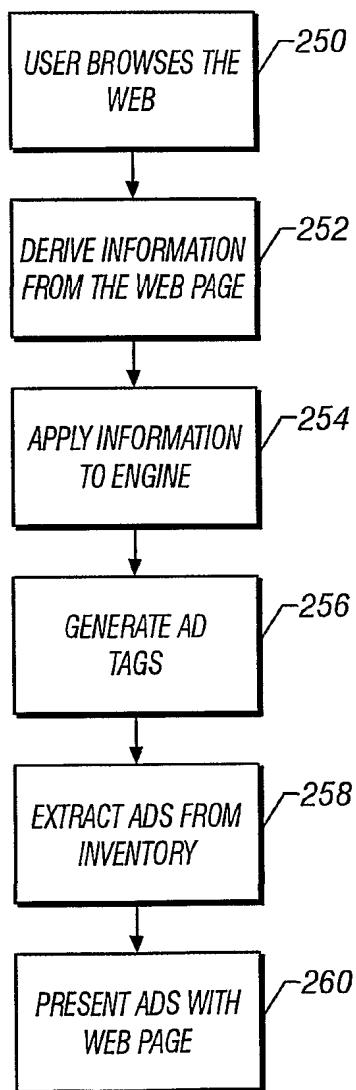


FIG. 7

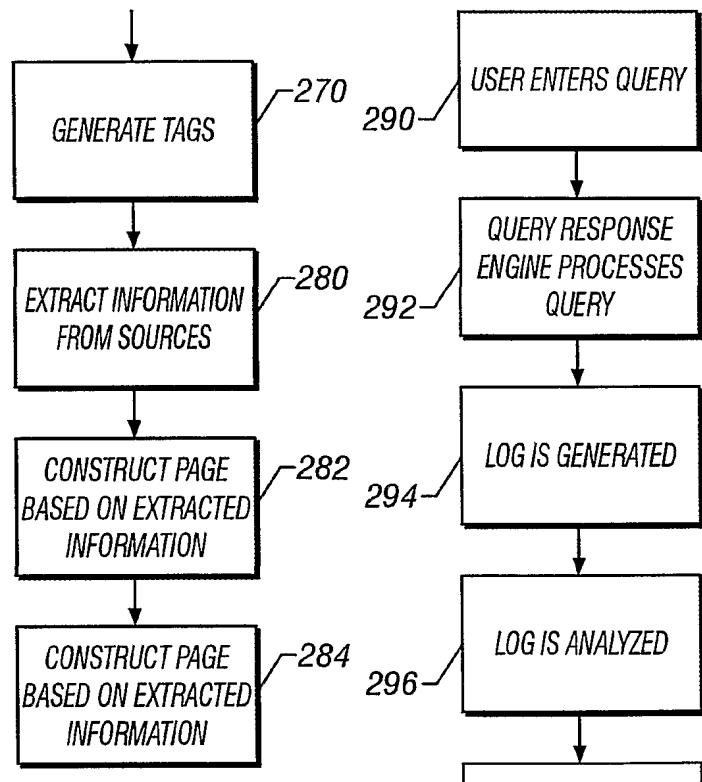


FIG. 8

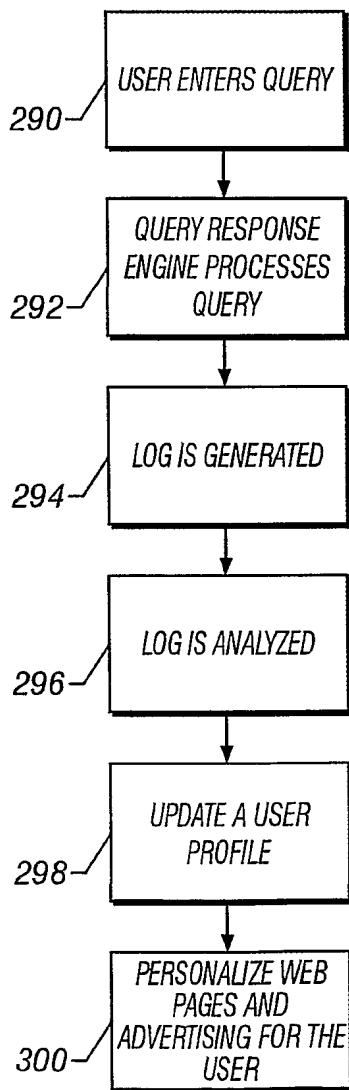
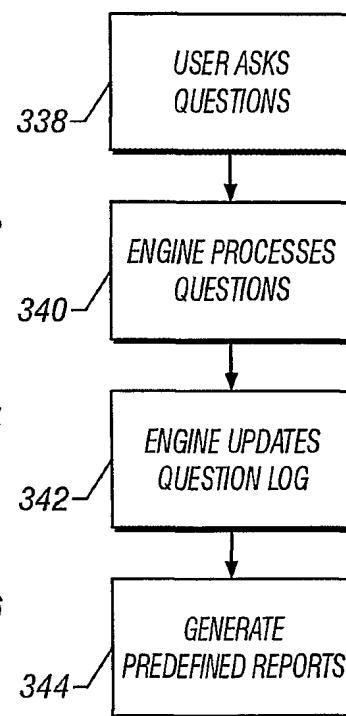
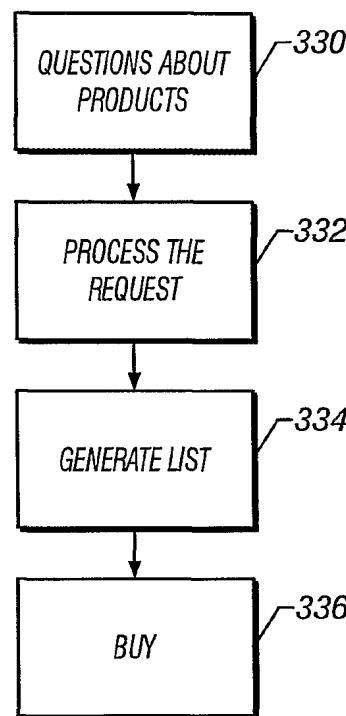
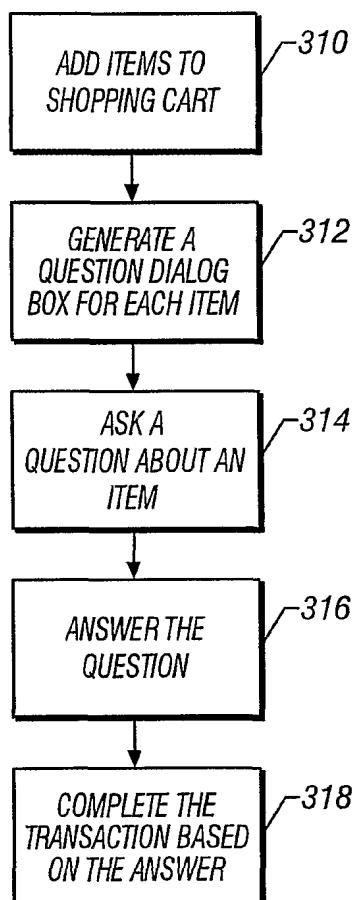


FIG. 9

6/8



7/8

David Feigenbaum 18

To: cia-answer@answerfriend.com 17

Subject: - What is the population of belgium?

12 16 { 10

14

FIG. 13

< b > Your Question: What is the population of Belgium? < /b > < /p >

Page 1 of 1

David Feigenbaum 44

From: cia-answer@answerfriend.com

Sent Thursday, August 03, 2000 8:13 AM

Subject: RE: What is the population of Belgium? 10

Population of Belgium: 10,182,034 (July 1999 est.) } 42
28

For more information about our services please see <http://www.answerfriend.com>

FIG. 14

8/8

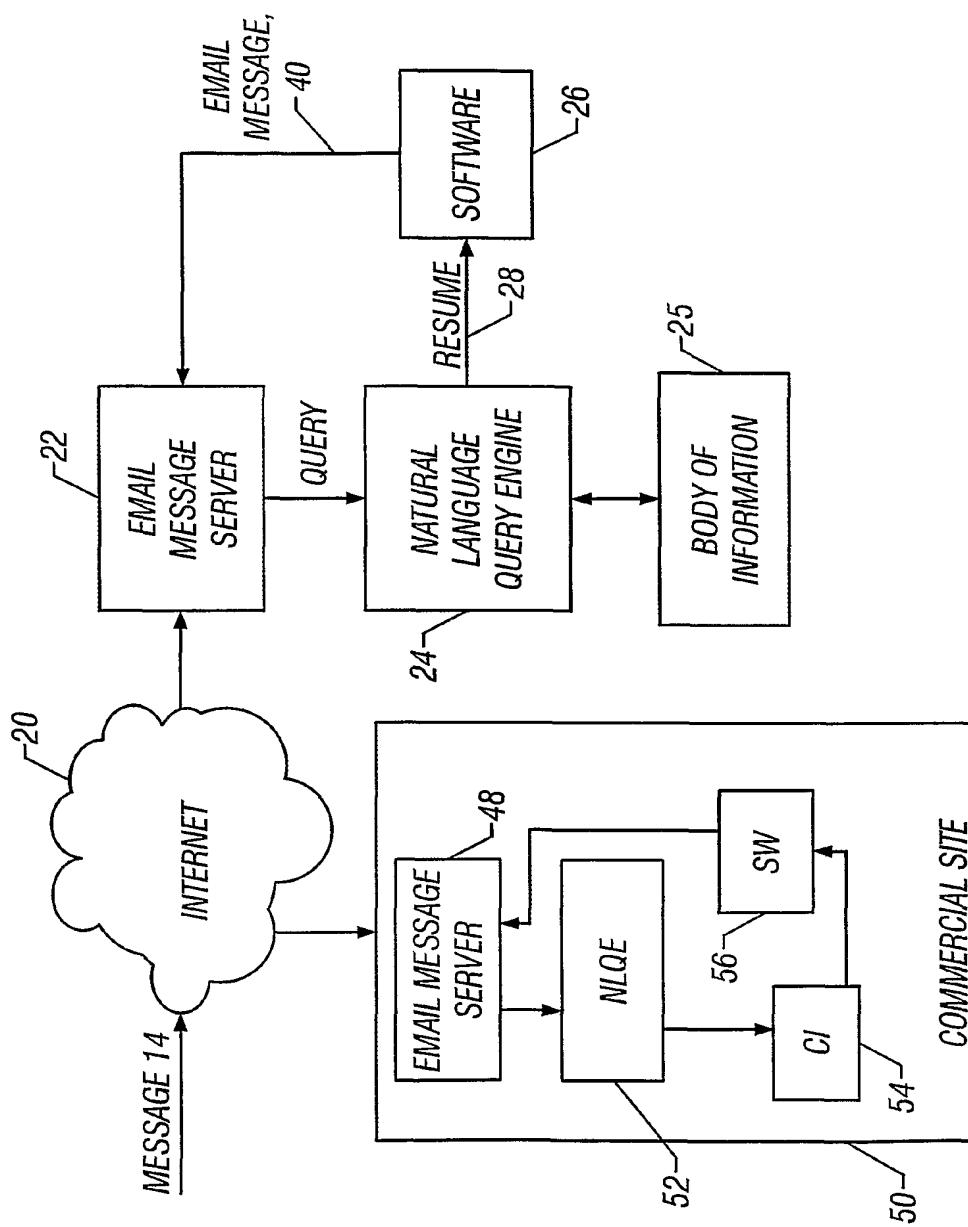


FIG. 15